CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being transmitted to Examiner Michael Young-Won via the USPTO EFS-Web on **November 17, 2006.**

Irena R. Nikolova

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of: Date: November 17, 2006

Michael J. JOHNSON Confirmation No: 7028

Serial No: 09/503,676 Group Art Unit: 2155

Filed: February 14, 2000 Examiner: Michael Young WON

For: GENERIC NETWORK PROTOCOL LAYER WITH SUPPORTING DATA STRUCTURE

SUBSTITUTE APPEAL BRIEF

Joseph A. Sawyer, Jr.
Attorney for Appellant
Sawyer Law Group, LLP
2465 E. Bayshore Road, Suite 406
Palo Alto, CA 94303

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

Appellant respectfully submits that International Business Machines Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellant states that no such proceeding exists.

III. STATUS OF CLAIMS

Claims 1-27 and 29-47 are pending and stand rejected. Accordingly, claims 1-27 and 29-47 are on appeal and all applied rejections concerning those claims are herein being appealed.

IV. STATUS OF AMENDMENT

Application Serial No. 09/503,676 (the instant application) as originally filed included claims 1-28. Claim 28 had been canceled and claims 29-47 had been added. Claims 1-27 and 29-47 are pending. Claims 1-27 and 29-47 are on appeal and all applied prospective rejections concerning claims 1-27 and 29-47 are being appealed herein. All amendments made to the instant application have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention provides a method for allowing the sharing of code between communications protocol layers and for eliminating the need for copying payload data between protocol layers in a sending system before the payload data is sent and between protocol layers in a receiving system after the payload data is received. A table describing seven standardized protocol layers (the "OSI Model") can be found in the specification beginning on page 2, line 11. The method of the present invention is accomplished by the creation of a generic protocol layer class (GPLC) having send and receive procedures and a common data buffer in which sent or received payload data is placed and acted upon by each protocol layer implemented with the GPLC. Instead of copying the payload data passed to it by a higher or lower protocol layer, a protocol layer acts upon the common data buffer by moving a "start" pointer and an "end" pointer along the data contained in the common data buffer prior to invoking the next higher or lower protocol layer. A first header and a second header are stored in the common data buffer at a location preceding the first start pointer, where the second header is contiguous with the first header. Each protocol layer can then process the payload data without it having to be copied. This reduces the amount of memory and processing time required for processing the payload data. (Summary, Figure 9, and page 15, lines 3-9.)

Independent claim 1 recites: In a network having a sending computer system and a receiving computer system, each of the sending and receiving computer systems including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for sending and receiving payload data by layers or sub-layers of at least one communications protocol, comprising the steps of:

(a) storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

(b) adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

(c) adjusting the first start pointer to point to a first byte of the first header;

(d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

(e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

(f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header;

(g) sending the payload data and the first and second headers to the receiving computer system;

(h) storing the payload data and the first and second headers in a second common data buffer of the receiving computer system;

(i) invoking a receive procedure of a second protocol layer of the communications protocol at the receiving computer system;

(j) storing a second start pointer pointing to a first byte of the second header in the second common data buffer;

(k) adjusting the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system;

(l) invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and

(m) transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

Independent claim 2 recites: In a computer system including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for sending payload data by layers or sub-layers of at least one communications protocol, the method comprising the steps of:

(a) storing a start pointer pointing to a first byte of the payload data in the common data buffer of the computer system;

(b) adding a first header to the payload data in the common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol;

(c) adjusting the start pointer to point to a first byte of the first header;

(d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

(e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure; and

(f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header.

Independent claim 6 recites: In a computer system including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for receiving payload data by layers or sub-layers of at least one communications protocol, the method comprising the steps of:

(a) storing the payload data, a first header, and a second header in the common data buffer of the receiving computer system, wherein the second header is contiguous with the first header;

(b) invoking a receive procedure of a second protocol layer of the communications protocol;

(c) storing a start pointer and an end pointer to the payload data;

(d) storing a second start pointer pointing to a first byte of the second header in the common data buffer;

(e) adjusting the start pointer to point to the first byte of the first header according to the second protocol layer;

(f) invoking a receive procedure of a first and higher protocol layer of the communications protocol; and

(g) transferring to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

Independent claim 10 recites: A computer system for sending and receiving payload data by layers or sub-layers of at least one communications protocol, the computer system comprising:

a processor for processing data from an application program;

a sending component for sending the payload data,

wherein the sending component stores a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

wherein the sending component adds a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

wherein the sending component adjusts the first start pointer to point to a first byte of the first header;

wherein the sending component invokes a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system; and

wherein the sending component transfers to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

wherein the sending component adds a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header; and

wherein the sending component sends the payload data and the first and second headers to the receiving computer system; and

a receiving component for receiving the payload data,

wherein the receiving component stores the payload data, the first header, and the second header in a second common data buffer of the receiving computer system;

wherein the receiving component invokes a receive procedure of a second protocol layer of the communications protocol at the receiving computer system;

wherein the receiving component adjusts the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system;

wherein the receiving component invokes a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and

wherein the receiving component transfers to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

Independent claim 11 recites: A computer system for sending payload data by layers or sub-layers of at least one communications protocol, the computer system comprising:

a processor for processing data from an application program;

a sending component for sending the payload data stored,

wherein the sending component stores a start pointer pointing to a first byte of the payload data in a common data buffer of the computer system;

wherein the sending component adds a first header to the payload data in the common data buffer at a location preceding the byte pointed to by the start pointer according to a first protocol layer of the communications protocol;

wherein the sending component adjusts the start pointer to point to a first byte of the first header;

wherein the sending component invokes a send procedure of a second and lower protocol layer of the communications protocol;

wherein the sending component transfers to the second protocol layer the start

pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

wherein the sending component adds a second header to the payload data in the common data buffer at a location preceding the start pointer, wherein the second header is contiguous with the first header; and

wherein the sending component sends the payload data and the first and second headers to a receiving computer system.

Independent claim 15 recites: A computer system for receiving payload data by layers or sub-layers of at least one communications protocol, the method comprising the steps of:

a processor for processing data from an application program;

a receiving component for receiving the payload data,

wherein the receiving component stores the payload data, a first header, and a second header in a common data buffer of the computer system, wherein the second header is contiguous with the first header;

wherein the receiving component invokes a receive procedure of a second protocol layer of the communications protocol;

wherein the sending component stores a start pointer and an end pointer to the payload data;

wherein the receiving component stores a second start pointer pointing to a first byte of the second header in the common data buffer;

wherein the receiving component adjusts the start pointer to point to the first byte of the first header according to the second protocol layer;

wherein the receiving component invokes a receive procedure of a first and higher protocol layer of the communications protocol; and

wherein the receiving component transfers to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

Independent claim 19 recites: A computer readable medium containing a computer program for the sending and receiving payload data by layers or sub-layers of at least one communications protocol, the computer program comprising program instructions for:

storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

adjusting the first start pointer to point to a first byte of the first header;

invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header;

sending the payload data and the first and second headers to the receiving computer system;

adjusting the second start pointer to point to the first byte of the first header according to

the second protocol layer at the receiving computer system;

invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and

transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

Independent claim 20 recites: A computer readable medium containing a computer program for sending payload data by layers or sub-layers of at least one communications protocol, the computer program comprising program instructions for:

storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

adjusting the first start pointer to point to a first byte of the first header;

invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header; and

sending the payload data and the first and second headers to the receiving computer

system.

Independent claim 24 recites: A computer readable medium containing a computer program for receiving payload data by layers or sub-layers of at least one communications protocol, the computer program comprising program instructions for:

storing the payload data, a first header, and a second header in the common data buffer of the receiving computer system, wherein the second header is contiguous with the first header;

invoking a receive procedure of a second protocol layer of the communications protocol;

storing a start pointer and an end pointer to the payload data;

storing a second start pointer pointing to a first byte of the second header in the common data buffer;

adjusting the start pointer to point to the first byte of the first header according to the second protocol layer;

invoking a receive procedure of a first and higher protocol layer of the communications protocol; and

transferring to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

Independent claim 29 recites: A method for processing payload data in a computer system using layers of a network communications protocol, the method comprising the steps of:

(a) storing the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header;

(b) processing the payload data using a first protocol layer of the network

communications protocol; and

(c) processing the payload data using a second protocol layer of the network communications protocol, wherein the payload data is not copied during and between being processed by the first and second protocol layers.

Independent claim 38 recites: A system for processing payload data using layers of a network communications protocol, the system comprising:

a processor for processing data from an application program; and

a component that stores the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header, wherein the component further processes the payload data using a first protocol layer of the network communications protocol and a second protocol layer of the network communications protocol, and wherein the payload data is not copied during and between being processed by the first and second protocol layers.

Independent claim 39 recites: A computer readable medium containing a computer program for processing payload data using layers of a network communications protocol, the computer program comprising program instructions for:

storing the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header;

processing the payload data using a first protocol layer of the network communications protocol; and

processing the payload data using a second protocol layer of the network communications

protocol, wherein the payload data is not copied during and between being processed by the first and second protocol layers.

Support for independent claims 1, 2, 6, 10-11, 15, 19-20, 24, 29, and 38-39 is found in the combination of the original claims; Figures 1 and 2; page 10, line 10, to page 12, line 13; Figures 3-11; on page 13, line 8, to page 15, line 21; and on page 5, lines 16-18.

More specifically, referring to independent claim 1, support for (a) "storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for (b) "adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 14, page 11, lines 1-2, Figure 4, element 160, and page 13, lines 16-18.

Support for (c) "adjusting the first start pointer to point to a first byte of the first header" is found in Figure 4, element 140, and page 13, lines 16-18.

Support for (d) "invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 16, and page 11, lines 2-3.

Support for (e) "transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure" is found in page 11, line 3, and page 5, lines 16-18.

Support for (f) "adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with

the first header" is found in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Support for (g) "sending the payload data and the first and second headers to the receiving computer system" is found in Figure 1, element 26, and page 11, lines 10-12.

Support for (h) "storing the payload data and the first and second headers in a second common data buffer of the receiving computer system" is found in Figure 11, element 550, and page 11, lines 10-12.

Support for (i) "invoking a receive procedure of a second protocol layer of the communications protocol at the receiving computer system" is found in Figure 2, element 52, and page 11, lines 17, to page 12, line 2.

Support for (j) "storing a second start pointer pointing to a first byte of the second header in the second common data buffer" is found in Figure 9, element 580, and page 15, lines 6-8.

Support for (k) "adjusting the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system" is found in Figure 10, element 580, and page 15, lines 11-13.

Support for (l) "invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system" is found in Figure 2, element 62, and page 12, lines 2-3.

Support for (m) "transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure" is found in page 12, line 3, and page 5, lines 16-18.

The features of claims 2, 6, 10-11, 15, 19-20, 24, 29, and 38-39 include various combinations of the limitations of claim 1. Accordingly, support for the limitations of claim 1 described above provide support for the limitations of claims 1, 2, 6, 10-11, 15, 19-20, 24, 29,

and 38-39.

More specifically, referring to independent claim 2, support for "(a) storing a start pointer pointing to a first byte of the payload data in the common data buffer of the computer system" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for "(b) adding a first header to the payload data in the common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol" is found in Figure 1, element 14, page 11, lines 1-2, Figure 4, element 160, and page 13, lines 16-18.

Support for "(c) adjusting the start pointer to point to a first byte of the first header" is found in Figure 4, element 140, and page 13, lines 16-18.

Support for "(d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 16, and page 11, lines 2-3.

Support for "(e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure" is found in page 11, line 3, and page 5, lines 16-18.

Support for "(f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header" is found in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Referring to independent claim 6, support for "(a) storing the payload data, a first header, and a second header in the common data buffer of the receiving computer system, wherein the

second header is contiguous with the first header" is found in Figure 11, element 550, and page 11, lines 10-12.

Support for "(b) invoking a receive procedure of a second protocol layer of the communications protocol" is found in Figure 2, element 52, and page 11, lines 17, to page 12, line 2.

Support for "(c) storing a start pointer and an end pointer to the payload data" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for "(d) storing a second start pointer pointing to a first byte of the second header in the common data buffer" is found in Figure 9, element 580, and page 15, lines 6-8.

Support for "(e) adjusting the start pointer to point to the first byte of the first header according to the second protocol layer" is found in Figure 10, element 580, and page 15, lines 11-13.

Support for "(f) invoking a receive procedure of a first and higher protocol layer of the communications protocol" is found in Figure 2, element 62, and page 12, lines 2-3.

Support for "(g) transferring to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure" is found in page 12, line 3, and page 5, lines 16-18.

Referring to independent claim 10, support for "a processor for processing data from an application program" is found in Figure 1, elements 10 and 20, and page 10, lines 10-19.

Support for "a sending component for sending the payload data" is found in Figure 1, elements 10, and page 10, line 10, to page 11, line 16.

Support for "wherein the sending component stores a first start pointer pointing to a first

byte of the payload data in a first common data buffer of the sending computer system" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for "wherein the sending component adds a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 14, page 11, lines 1-2, Figure 4, element 160, and page 13, lines 16-18.

Support for "wherein the sending component adjusts the first start pointer to point to a first byte of the first header" is found in Figure 4, element 140, and page 13, lines 16-18.

Support for "wherein the sending component invokes a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 16, and page 11, lines 2-3.

Support for "wherein the sending component transfers to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure" is found in page 11, line 3, and page 5, lines 16-18.

Support for "wherein the sending component adds a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header" is found in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Support for "wherein the sending component sends the payload data and the first and second headers to the receiving computer system" is found in Figure 1, element 26, and page 11, lines 10-12.

Support for "a receiving component for receiving the payload data" is found in Figure 1,

element 20, and page 10, line 10, to page 11, line 16.

Support for "wherein the receiving component stores the payload data, the first header, and the second header in a second common data buffer of the receiving computer system" is found in Figure 11, element 550, and page 11, lines 10-12.

Support for "wherein the receiving component invokes a receive procedure of a second protocol layer of the communications protocol at the receiving computer system" is found in Figure 2, element 52, and page 11, lines 17, to page 12, line 2.

Support for "wherein the receiving component adjusts the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system" is found in Figure 10, element 580, and page 15, lines 11-13.

Support for "wherein the receiving component invokes a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system" is found in Figure 2, element 62, and page 12, lines 2-3.

Support for "wherein the receiving component transfers to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure" is found in page 12, line 3, and page 5, lines 16-18.

Referring to independent claim 11, support for "a processor for processing data from an application program" is found in Figure 1, elements 10 and 20, and page 10, lines 10-19.

Support for "a sending component for sending the payload data stored" is found in Figure 1, element 10, and page 10, line 10, to page 11, line 16.

Support for "wherein the sending component stores a start pointer pointing to a first byte

of the payload data in a common data buffer of the computer system" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for "wherein the sending component adds a first header to the payload data in the common data buffer at a location preceding the byte pointed to by the start pointer according to a first protocol layer of the communications protocol" is found in Figure 1, element 14, page 11, lines 1-2, Figure 4, element 160, and page 13, lines 16-18.

Support for "wherein the sending component adjusts the start pointer to point to a first byte of the first header" is found in Figure 4, element 140, and page 13, lines 16-18.

Support for "wherein the sending component invokes a send procedure of a second and lower protocol layer of the communications protocol" is found in Figure 1, element 16, and page 11, lines 2-3.

Support for "wherein the sending component transfers to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure" is found in page 11, line 3, and page 5, lines 16-18.

Support for "wherein the sending component adds a second header to the payload data in the common data buffer at a location preceding the start pointer, wherein the second header is contiguous with the first header" is found in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Support for "wherein the sending component sends the payload data and the first and second headers to a receiving computer system" is found in Figure 1, element 26, and page 11, lines 10-12.

Referring to independent claim 15, support for "a processor for processing data from an

application program" is found in Figure 1, elements 10 and 20, and page 10, lines 10-19.

Support for "a receiving component for receiving the payload data" is found in Figure 1, elements 20, and page 10, line 10, to page 11, line 16.

Support for "wherein the receiving component stores the payload data, a first header, and a second header in a common data buffer of the computer system, wherein the second header is contiguous with the first header" is found in Figure 1, elements 10 and 20, and page 10, lines 10-19, Figure 5, element 160, Figure 9, element 580, and page 15, lines 6-8.

Support for "wherein the receiving component invokes a receive procedure of a second protocol layer of the communications protocol" is found in Figure 2, element 52, and page 11, lines 17, to page 12, line 2.

Support for "wherein the sending component stores a start pointer and an end pointer to the payload data" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for "wherein the receiving component stores a second start pointer pointing to a first byte of the second header in the common data buffer" is found in Figure 9, element 580, and page 15, lines 6-8.

Support for "wherein the receiving component adjusts the start pointer to point to the first byte of the first header according to the second protocol layer" is found in Figure 10, element 580, and page 15, lines 11-13.

Support for "wherein the receiving component invokes a receive procedure of a first and higher protocol layer of the communications protocol" is found in Figure 2, element 62, and page 12, lines 2-3.

Support for "wherein the receiving component transfers to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive

procedure" is found in page 12, line 3, and page 5, lines 16-18.

Referring to independent claim 19, support for "storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

Support for "adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 14, page 11, lines 1-2, Figure 4, element 160, and page 13, lines 16-18.

Support for "adjusting the first start pointer to point to a first byte of the first header" is found in Figure 4, element 140, and page 13, lines 16-18.

Support for "invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 16, and page 11, lines 2-3.

Support for "transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure" is found in page 11, line 3, and page 5, lines 16-18.

Support for "adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header" is found in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Support for "sending the payload data and the first and second headers to the receiving computer system" is found in Figure 1, element 26, and page 11, lines 10-12.

Support for "adjusting the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system" is found in Figure 10,

element 580, and page 15, lines 11-13.

Support for "invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system" is found in Figure 2, element 62, and page 12, lines 2-3.

Support for "transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure" is found in page 12, line 3, and page 5, lines 16-18.

Referring to independent claim 20, support for "storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

Support for "adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 14, page 11, lines 1-2, Figure 4, element 160, and page 13, lines 16-18.

Support for "adjusting the first start pointer to point to a first byte of the first header" is found in Figure 4, element 140, and page 13, lines 16-18.

Support for "invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system" is found in Figure 1, element 16, and page 11, lines 2-3.

Support for "transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure" is found in page 11, line 3, and page 5, lines 16-18.

Support for "adding a second header to the payload data in the first common data buffer

at a location preceding the first start pointer, wherein the second header is contiguous with the first header" is found in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Support for "sending the payload data and the first and second headers to the receiving computer system" is found in Figure 1, element 26, and page 11, lines 10-12.

Referring to independent claim 24, support for "storing the payload data, a first header, and a second header in the common data buffer of the receiving computer system, wherein the second header is contiguous with the first header" is found in Figure 11, element 550, and page 11, lines 10-12.

Support for "invoking a receive procedure of a second protocol layer of the communications protocol" is found in Figure 2, element 52, and page 11, lines 17, to page 12, line 2.

Support for "storing a start pointer and an end pointer to the payload data" is found in Figure 3, element 140, and page 13, lines 12-13.

Support for "storing a second start pointer pointing to a first byte of the second header in the common data buffer" is found in Figure 9, element 580, and page 15, lines 6-8.

Support for "adjusting the start pointer to point to the first byte of the first header according to the second protocol layer" is found in Figure 10, element 580, and page 15, lines 11-13.

Support for "invoking a receive procedure of a first and higher protocol layer of the communications protocol" is found in Figure 2, element 62, and page 12, lines 2-3.

Support for "transferring to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure" is found in page 12, line 3,

and page 5, lines 16-18.

Referring to independent claim 29, support for "(a) storing the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header" is found in Figure 1, elements 10 and 20, and page 10, lines 10-19, Figure 5, element 160, Figure 9, element 580, and page 15, lines 6-8.

Support for "(b) processing the payload data using a first protocol layer of the network communications protocol" is found in Figure 1, element 24, page 11, lines 9-10.

Support for "(c) processing the payload data using a second protocol layer of the network communications protocol, wherein the payload data is not copied during and between being processed by the first and second protocol layers" is found in page 12, line 3, and page 5, lines 16-18.

Referring to independent claim 38, support for "a processor for processing data from an application program" is found in Figure 1, elements 10 and 20, page 10, lines 10-19.

Support for "a component that stores the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header, wherein the component further processes the payload data using a first protocol layer of the network communications protocol and a second protocol layer of the network communications protocol, and wherein the payload data is not copied during and between being processed by the first and second protocol layers" is found in Figure 9, element 580, page 15, lines 6-8, in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160, and in page 12, line 3, and page 5, lines 16-18.

Referring to independent claim 39, support for "storing the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header

Support for "processing the payload data using a first protocol layer of the network communications protocol" is found in Figure 1, elements 10 and 20, lines 10-19, and in Figure 1, element 24, page 11, lines 9-10, and Figure 5, element 160.

Support for "processing the payload data using a second protocol layer of the network communications protocol, wherein the payload data is not copied during and between being processed by the first and second protocol layers" is found in Figure 1, elements 10 and 20, lines 10-19, and page 12, line 3, and page 5, lines 16-18.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellant respectfully seeks review of the following rejections:

1. Claims 1, 2, 4-6, 9-11, 13-15, 18-20, 22-24, 27, 29-34, 36, 38-44, and 46 are rejected under 35 U.S.C. 102(e) as being anticipated by Wilson et al. (U.S. Patent No. 6,651,117 B1).

2. Claims 3, 7, 8, 12, 16, 17, 21, 25, 26, 35, 37, 45, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilson in view of Boucher et al. (U.S. Patent No. 6,226,680 B1).

VII. ARGUMENTS

A. Summary of the Applied Rejections

The Final Office Action dated June 17, 2005 rejected claims 1, 2, 4-6, 9-11, 13-15, 18-20, 22-24, 27, 29-34, 36, 38-44, and 46 under 35 U.S.C. 102(e) as being anticipated by Wilson et al. (U.S. Patent No. 6,651,117 B1). In making the rejection, the Examiner stated:

> **Claims 1, 2, 4-6, 9-11, 13-15, 18-20, 22-24, 27, 29-34, 36, 38-44, and 46 are rejected under 35 U.S.C. 102(e) as being anticipated by Wilson et al. (US 6,651,117 B1).**
> **INDEPENDENT:**
>
> **As per claims 1, 2, 6, 10, 11, 15, 19, 20, and 24, Wilson teaches of a network having a sending computer system and a receiving computer system, each of the sending and receiving computer systems including a processor, a memory (see col. 7, lines 24) and a network adapter (see col. 6, line 10: NIC), the memory containing a data structure used for storing a common data buffer (see col. 6, lines 44-48), a method, system, and a computer readable medium containing a computer program, for sending and receiving payload data by layers or sub-layers (see Fig. 1) of at least one communications protocol (see abstract), the method comprising the steps of: (a) storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system (see col. 6, lines 62-67); (b) adding a first header (see col.3, line 59-60) to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system (see col.6, line 67 to col.7, line 4); (c) adjusting the first start pointer to point t a first byte of the first header (see col. 6, lines 5-8); (d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system (see col.6, lines 62-64 and col.8, lines 34-35); (e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure (see col.2, lines 59-61; col.3, lines 51-58; and col.6, lines 1-8); (f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer (see col.6, line 67 to col.7, line 4) (NOTE: Wilson teaches essentially that prior art network layer communication is allowed (see col.3, lines 51-52), thus the number of protocol layers does not functionally distinguish the invention and furthermore, after each succession of layers the pointers are clearly adjusted (see col.6, lines 5-8)), wherein the second header is contiguous with the first header (see Fig. 3 and col. 6, lines 31-33 & 36-38); (g) sending the payload data and the first and second headers to the receiving computer system (see col.8, lines 34-35); (h) storing the payload data and the first and second headers in a second common data buffer of the receiving computer system (see col.9, lines 10-12))), wherein the second header is contiguous with the first header (see Fig. 3 and col. 6, lines 31-33 & 36-38); (i) invoking a receive procedure of a second protocol layer of the communications protocol at the receiving computer system (see col.9, lines 19-24); (j) storing a pointer and end**

pointer to the payload data and also storing a second start pointer pointing to a first byte of the second header in the second common data buffer (see col.9, lines 45-48); (k) adjusting a the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system (see col.6, lines 5-8); (l) invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system (see note above and col.9, lines 19-24); and (m) transferring to the first protocol layer at the receiving computer system the second start pointer (see note above), wherein the payload data is not copied in preparation for or during the receive procedure (see col.2, lines 59-61; col.3, lines 51-58; and col.6, lines 1-8.

The Final Office Action dated June 17, 2005 also rejected claims 3, 7, 8, 12, 16, 17, 21, 25, 26, 35, 37, 45, and 47 under 35 U.S.C. 103(a) as being unpatentable over Wilson in view of Boucher et al. (U.S. Patent No. 6226680 B1).

The Examiner stated the following in response to the previous arguments against the 102(e) rejections:

Applicant's arguments filed December 20, 2004 have been fully considered but they are not persuasive. *Wilson* clearly teaches the element of the amended claims.

With respect to the argument presented by the applicant(s), the cited Figure 4 of the *Wilson* patent is describing a specific situation wherein the data packet to be transmitted is too large to transmit as a single packet and must be broken up into smaller "chunks" (see col. 7, lines 38-50). Such technique is well known in the art wherein a header precedes the payload portions for successfully re-joining the chunks at the receiving device. The functionality taught by *Wilson* does not specifically apply to the recited broad limitation of the claimed invention because the claimed invention does not recite wherein the data is transmitted in partitioned chunks:

***Wilson* teaches in column 6, lines 31-33 & 36-38 with reference to Fig. 3, that the SID header portion includes a common SID header and a layer specific SID header, which clearly teaches of a "second header contiguous with a first header".**

Furthermore, *Wilson* teaches of overcoming conventional prior art technique of buffer copying as data is processed from one layer to the next (see col. 2, lines 25-56). Such prior art technique is well known wherein the header of each layer is appended next to the previous header (contiguous). *Wilson* teaches of an interface that is "essentially standardized, thus allowing similar routines to use the interface" (see col. 2, lines 62-63). The main distinction of *Wilson* patent over prior art is replacing buffer descriptors including memory address pointers (see col. 3, lines 11-12) in place of payload data to "greatly reducing the amount of copying performed during inter-layer communication" (see col. 3, lines 2-5).

The Examiner stated the following in the Advisory Action dated June 7, 2005:

In response to the description of the claimed invention recited on the first paragraph of page 22 of the Response After Final filed August 19, 2005, Wilson et al. US Pat. No. 6,651,117 (hereafter "Wilson") also teaches of the seven standardized protocol layers (the "OSI Model") as prior art. Although Wilson does not explicitly state a generic protocol layer class (GPLC), Wilson clearly teaches and suggests such means (see col. 3, lines 2-15). Similarly Wilson teaches rather than copying payload data passed to it by the different protocol layers (see col. 3, lines 51-58), the protocol layers act upon the data by moving pointers along the data (see col. 3, lines 2-5 and lines 18-32; and Fig. 4, #142, #144, #146: 1st Buffer Descriptor, 2nd Buffer Descriptor, and 3rd Buffer Descriptor). In Figure 4, Wilson teaches of these descriptors appending to the head of the 4K buffer. In essence, these buffer descriptors are exactly the headers described in the claimed invention and as taught by Figure 4, are clearly "contiguous". It is also inherent that these buffer descriptors are layer specific because Wilson teaches during network communication information travels via the different layers using a network stack interface called SCSI Interface Descriptor (SID) (see col. 5, lines 51-67). Furthermore, Wilson teaches the inter-layer data transfer is occurs by passing memory address pointers via SIDs (see col. 6, lines 1-8).

Clearly the above-cited references in combination with the cited reference locations of the Final Office action explicitly teach, "wherein the second header is contiguous with the first header".

In response to the non-applicable functionality taught by Wilson, just because Wilson teaches a special circumstance regarding partitioning data into chunks due to buffers that are too large to transmit as a single packet, does not mean that the art is non-analogous. Similarly, if a reference teaches additional limitations and/or utility it does not make the reference non-analogous. The applicant is reminded that the reference is not in question as to its patentability. Even if the data in the buffer is too large to transmit as a single packet, Wilson clearly teaches that the headers (Descriptors) are contiguous (see Fig. 4, #126, #142, #144, and #146: the contiguous headers are appended to the head of the first partition, #148, of the 4K Buffer).

In response to the argument regarding "transferring to the second protocol layer the starting pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure", Wilson clearly teaches this limitation. The reference to column 2 clearly does not teach this limitation because column 2 is conventional prior art. Wilson teaches in column 3, lines 54-58 that "data is obtained from the packet buffer without performing copy operation for each network stack layer". Such teachings are consistent through the Wilson patent.

For the reasons above, claims 2, 6, 10-11, 15, 19-20, 24, 29, and 38-39 are similarly rejected and therefore, the dependent claims remain rejected.

Additionally, the examiner could not conclude that the "contiguous" aspect of the claimed invention is the novel element of the claimed invention because such terminology in never mentioned in the disclosure.

Appellants respectfully request that the Board reverse the Examiner's final rejection of the pending claims.

B. The Cited Prior Art

Wilson discloses a network stack layer interface for communication between network stack layers. The network stack layer interface includes a header portion that defines various characteristics of the network stack layer interface. In addition, a buffer descriptor is included that defines data that was, or will be, transmitted over a computer network. The buffer descriptor includes a memory address pointer to the data. In this manner, information is passed between network stack layers via the network stack interface, resulting in fast network data transfer with reduced data copying. (Abstract.)

Boucher discloses a system for protocol processing in a computer network and includes an intelligent network interface card, which provides a fast-path that avoids protocol processing for messages, greatly accelerating data communication (Abstract).

C. Claims 1-27 and 29-47 Are Not Unpatentable Under 35 U.S.C. 102(e) and 103(a)

The present invention provides a method for allowing the sharing of code between communications protocol layers and for eliminating the need for copying payload data between protocol layers in a sending system before the payload data is sent and between protocol layers in a receiving system after the payload data is received. A table describing seven standardized protocol layers (the "OSI Model") can be found in the specification beginning on page 2, line 11. The method of the present invention is accomplished by the creation of a generic protocol layer class (GPLC) having send and receive procedures and a common data buffer in which sent or received payload data is placed and acted upon by each protocol layer implemented with the GPLC. Instead of copying the payload data passed to it by a higher or lower protocol layer, a protocol layer acts upon the common data buffer by moving a "start" pointer and an "end" pointer along the data contained in the common data buffer prior to invoking the next higher or lower protocol layer. A first header and a second header are stored in the common data buffer at a location preceding the first start pointer, where the second header is contiguous with the first header. Each protocol layer can then process the payload data without it having to be copied.

This reduces the amount of memory and processing time required for processing the payload data. (Summary, Figure 9, and page 15, lines 3-9.) Wilson does not teach or suggest these features, as discussed below.

Wilson does not teach or suggest "storing a first header and a second header to the payload data in a common data buffer, wherein the second header is contiguous with the first header," as recited in independent claim 1. The Examiner has referred to column 6, lines 31-33 and 36-38, of Wilson as teaching this feature. However, these sections of Wilson merely state that the SID header portion preferably "includes a common SID header and a layer specific SID header" (column 6, lines 31-33) and that "the layer specific SID header includes data that is particularly useful for the specific layer utilizing the SID" (column 6, lines 36-38). Nowhere do these sections specifically describe the structure of the SID header, and these sections clearly do not specifically describe that the common SID header and the layer specific SID headers are contiguous. Furthermore, nowhere do these sections teach or suggest the rest of the feature, "adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer," as recited in the present invention.

With regard to the teaching of Wilson where multiple non-contiguous headers are described (Figure 4 and column 7, lines 38-50), the Examiner has stated that the functionality of Wilson does not specifically apply to the claimed invention because the claimed invention does not recite that the data is transmitted in partitioned chunks. However, Figure 4 is an "exemplary SID flow" that Wilson uses to transmit data in a network environment. The advantage of the present invention, where the headers are contiguous, is that only one start pointer is needed. Wilson does not provide this benefit since the headers of Wilson are not contiguous. Consequently, numerous pointers are required in Wilson to locate each header. If the functionality of Wilson does not specifically apply to the claimed invention, as suggested by the

Examiner, Applicant respectfully submits that Wilson should not be applied to the claimed invention, since Wilson does not teach or suggest "adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header," as recited in the present invention.

Furthermore, not only does Wilson not describe the above-described adding feature, including contiguous headers, but Wilson also does not teach or suggest "transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure," as recited in independent claim 1. The Examiner has suggested that Wilson overcomes conventional prior art of buffer copying, referring to column 2, lines 25-56, of Wilson. However, Wilson in this section clearly describes copying, where "copying is used to pass processed data from one network stack layer to the next." This section further states that "the entire data buffer must be copied to a new buffer each time a new network stack layer needs to access it" (column 2, lines 39-48). Wilson mentions in column 2, lines 62-63, an interface that "should be essentially standardized, thus allowing similar routines to use the interface," as suggested by the Examiner. However, this section does not teach or suggest that "payload data is not copied in preparation for or during send and receive procedures using protocol layers," as recited in the present invention. In fact, where the Examiner points out the main distinction of Wilson over the prior art (column 3, lines 2-5 and 54-58), Wilson merely describes that an interface is configured to pass memory address pointers, thus "greatly reducing" the amount of copying performed during inter-layer communication, and that information is passed between network stack layers via the network stack interface, resulting in fast network data transfer with "reduced data copying." Clearly, data copying still exists in Wilson and is merely "reduced." These descriptions strongly suggest that Wilson teaches away

from the present invention as claimed, where payload data is not copied in preparation for or during send and receive procedures using protocol layers.

The Examiner has referred to 1st buffer descriptor 142, 2nd buffer descriptor 144, and 3rd buffer descriptor 146 of Figure 4 of Wilson, stating that "these buffer descriptors are exactly the headers described in the claimed invention and as taught by Figure 4, are clearly "contiguous." However, Applicant respectfully submits that these buffer descriptors are merely "descriptors," which are attached to an SID header 126. As Figure 4 of Wilson clearly illustrates, SID header 126 is not contiguous with the other SID headers 122, 124, 128, and 130. In contrast to Wilson, referring to Figure 5 of the present invention, the headers 160 and 170 are clearly contiguous. Referring also to Figures 7 of the present invention, the headers 200 and 210 are clearly contiguous.

Therefore, Wilson not teach or suggest the combination of steps as recited in independent claim 1, and this claim is allowable over Wilson.

Independent claims 2, 6, 10-11, 15, 19-20, 24, 29, and 38-39 recite transferring, a start pointer, "wherein the payload data is not copied in preparation for or during the send procedure," and storing a first header and a second header to the payload data in a common data buffer, "wherein the second header is contiguous with the first header." As described above, with respect to independent claim 1, Wilson does not teach or suggest this feature. Accordingly, the above-articulated arguments related to independent claim 1 apply with equal force to claims 2, 6, 10-11, 15, 19-20, 24, 29, and 38-39. Therefore, these claims are allowable over the cited reference for at least the same reasons as claim 1.

With regard to the 103(a) rejections, Applicant respectfully disagrees with the Examiner's rejections. Dependent claims 3, 7, 8, 12, 16, 17, 21, 25, 26, 35, 37, 45, and 47 depend from

independent claims 2, 6, 11, 15, 24, 29, and 39, respectively. Accordingly, the above-articulated arguments related to independent claims 2, 6, 11, 15, 24, 29, and 39 apply with equal force to claims 3, 7, 8, 12, 16, 17, 21, 25, 26, 35, 37, 45, and 47, which are thus allowable over the cited reference for at least the same reasons as claims 2, 6, 11, 15, 24, 29, and 39.

With regard to the remaining dependent claims, dependent claims 4-5, 9, 13-14, 18, 22-23, 27, 30-34, 36, 40-44, and 46 dependent from independent claims 2, 6, 11, 15, 20, 24, 29, and 39, respectively. Accordingly, the above-articulated arguments related to independent claims 2, 6, 11, 15, 20, 24, 29, and 39 apply with equal force to claims 4-5, 9, 13-14, 18, 22-23, 27, 30-34, 36, 40-44, and 46, which are thus allowable over the cited references for at least the same reasons as claims 2, 6, 11, 15, 20, 24, 29, and 39.

In view of the foregoing, Applicant respectfully submits that the recited invention is not taught, shown, or suggested by the cited art.

Accordingly, Appellant respectfully requests withdrawal of the rejection under 35 U.S.C. 102 (e) and 103(a) and respectfully requests that the Board reverse the final rejection of claims.

D. Summary of Arguments

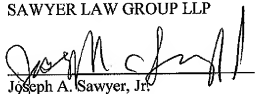
For all the foregoing reasons, it is respectfully submitted that claims 1-27 and 29-47 (all of the claims presently in the application) are patentable for defining subject matter, which would not have been unpatentable under 35 U.S.C. 102(e) and 103(a) at the time the subject matter was invented. Thus, Appellants respectfully request that the Board reverse the rejection of all the appealed claims and find each of these claims allowable.

Note: For convenience of detachment without disturbing the integrity of the remainder of pages of this Appeal Brief, Appellants' APPENDICES A-C are attached on separate sheets following the signatory portion of this Appeal Brief.

Please charge any fee that may be necessary for the continued pendency of this application to Deposit Account No. 50-0563 (IBM Corporation).

Respectfully submitted,

SAWYER LAW GROUP LLP



November 17, 2006
Date

Joseph A. Sawyer, Jr.
Attorney for Appellant
Reg. No. 30,801
(650) 493-4540

APPENDIX A

CLAIMS

1. (Previously presented) In a network having a sending computer system and a receiving computer system, each of the sending and receiving computer systems including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for sending and receiving payload data by layers or sub-layers of at least one communications protocol, comprising the steps of:

(a) storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

(b) adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

(c) adjusting the first start pointer to point to a first byte of the first header;

(d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

(e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

(f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header;

(g) sending the payload data and the first and second headers to the receiving computer system;

(h) storing the payload data and the first and second headers in a second common data buffer of the receiving computer system;

(i) invoking a receive procedure of a second protocol layer of the communications protocol at the receiving computer system;

(j) storing a second start pointer pointing to a first byte of the second header in the second common data buffer;

(k) adjusting the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system;

(l) invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and

(m) transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

2. (Previously presented) In a computer system including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for sending payload data by layers or sub-layers of at least one communications protocol, the method comprising the steps of:

(a) storing a start pointer pointing to a first byte of the payload data in the common data buffer of the computer system;

(b) adding a first header to the payload data in the common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol;

(c) adjusting the start pointer to point to a first byte of the first header;

(d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

(e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure; and

(f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header.

3. (Previously presented) The method of claim 2 wherein a checksum is added to the header in the common data buffer preceding the payload data being sent.

4. (Previously presented) The method of claim 2 wherein the transferring step includes any application data or information required by the send procedure of the second and lower protocol layer.

5. (Previously presented) The method of claim 2 further comprising the step of (f) adjusting a size of the payload data to be sent by the second and lower protocol layer by adjusting the end pointer.

6. (Previously presented) In a computer system including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for receiving payload data by layers or sub-layers of at least one communications

protocol, the method comprising the steps of:

(a) storing the payload data, a first header, and a second header in the common data buffer of the receiving computer system, wherein the second header is contiguous with the first header;

(b) invoking a receive procedure of a second protocol layer of the communications protocol;

(c) storing a start pointer and an end pointer to the payload data;

(d) storing a second start pointer pointing to a first byte of the second header in the common data buffer;

(e) adjusting the start pointer to point to the first byte of the first header according to the second protocol layer;

(f) invoking a receive procedure of a first and higher protocol layer of the communications protocol; and

(g) transferring to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

7. (Previously presented) The method of claim 6 wherein a checksum following the header and added by the sending computer system is removed from the received payload data in the common data buffer.

8. (Previously presented) The method of claim 7 wherein the checksum is removed by adjusting the start pointer of the common data buffer to point to a memory location following the checksum.

9. (Previously presented) The method of claim 6 further comprising the step of (m) transferring any application data or information required by the receive procedure of the first and higher protocol layer.

10. (Previously presented) A computer system for sending and receiving payload data by layers or sub-layers of at least one communications protocol, the computer system comprising:

a processor for processing data from an application program;

a sending component for sending the payload data,

wherein the sending component stores a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

wherein the sending component adds a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

wherein the sending component adjusts the first start pointer to point to a first byte of the first header;

wherein the sending component invokes a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system; and

wherein the sending component transfers to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

wherein the sending component adds a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header

is contiguous with the first header; and

wherein the sending component sends the payload data and the first and second headers to the receiving computer system; and

a receiving component for receiving the payload data,

wherein the receiving component stores the payload data, the first header, and the second header in a second common data buffer of the receiving computer system;

wherein the receiving component invokes a receive procedure of a second protocol layer of the communications protocol at the receiving computer system;

wherein the receiving component adjusts the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system;

wherein the receiving component invokes a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and

wherein the receiving component transfers to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

11. (Previously presented) A computer system for sending payload data by layers or sub-layers of at least one communications protocol, the computer system comprising:

a processor for processing data from an application program;

a sending component for sending the payload data stored,

wherein the sending component stores a start pointer pointing to a first byte of the payload data in a common data buffer of the computer system;

wherein the sending component adds a first header to the payload data in the common data buffer at a location preceding the byte pointed to by the start pointer according to a first protocol layer of the communications protocol;

wherein the sending component adjusts the start pointer to point to a first byte of the first header;

wherein the sending component invokes a send procedure of a second and lower protocol layer of the communications protocol;

wherein the sending component transfers to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

wherein the sending component adds a second header to the payload data in the common data buffer at a location preceding the start pointer, wherein the second header is contiguous with the first header; and

wherein the sending component sends the payload data and the first and second headers to a receiving computer system.

12. (Previously presented) The computer system of claim 11 wherein the sending component adds a checksum to the header in the common data buffer preceding the payload data being sent.

13. (Previously presented) The computer system of claim 11 wherein the sending component transfers any application data or information required by the send procedure of the second and lower protocol layer.

14. (Previously presented) The computer system of claim 11 wherein the sending component adjusts a size of data to be sent by the second and lower protocol layer by adjusting the end pointer.

15. (Previously presented) A computer system for receiving payload data by layers or sub-layers of at least one communications protocol, the method comprising the steps of:

a processor for processing data from an application program;

a receiving component for receiving the payload data,

wherein the receiving component stores the payload data, a first header, and a second header in a common data buffer of the computer system, wherein the second header is contiguous with the first header;

wherein the receiving component invokes a receive procedure of a second protocol layer of the communications protocol;

wherein the sending component stores a start pointer and an end pointer to the payload data;

wherein the receiving component stores a second start pointer pointing to a first byte of the second header in the common data buffer;

wherein the receiving component adjusts the start pointer to point to the first byte of the first header according to the second protocol layer;

wherein the receiving component invokes a receive procedure of a first and higher protocol layer of the communications protocol; and

wherein the receiving component transfers to the first protocol layer the start

pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

16. (Previously presented) The computer system of claim 15 wherein the receiving component removes a checksum added by the sending computer system from the received payload data in the common data buffer.

17. (Previously presented) The computer system of claim 16 wherein the receiving component removes the checksum by adjusting the start pointer of the common data buffer to point to a memory location following the checksum.

18. (Previously presented) The computer system of claim 15 wherein the receiving component transfers any application data or information required by the receive procedure of the first and higher protocol layer.

19. (Previously presented) A computer readable medium containing a computer program for the sending and receiving payload data by layers or sub-layers of at least one communications protocol, the computer program comprising program instructions for:

storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

adjusting the first start pointer to point to a first byte of the first header;

invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header;

sending the payload data and the first and second headers to the receiving computer system;

adjusting the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system;

invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and

transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

20. (Previously presented) A computer readable medium containing a computer program for sending payload data by layers or sub-layers of at least one communications protocol, the computer program comprising program instructions for:

storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;

adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;

adjusting the first start pointer to point to a first byte of the first header;

invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;

transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;

adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer, wherein the second header is contiguous with the first header; and

sending the payload data and the first and second headers to the receiving computer system.

21. (Previously presented) The computer readable medium of claim 20 wherein the computer instructions add a checksum to the header in the common data buffer preceding the payload data being sent.

22. (Previously presented) The computer readable medium of claim 20 wherein the computer instructions transfer any application data or information required by the send procedure of the second and lower protocol layer.

23. (Previously presented) The computer readable program medium of claim 20 wherein the computer instructions adjust a size of the payload data to be sent by the second and lower protocol layer by adjusting the end pointer.

24. (Previously presented) A computer readable medium containing a computer program for receiving payload data by layers or sub-layers of at least one communications protocol, the computer program comprising program instructions for:

storing the payload data, a first header, and a second header in the common data buffer of the receiving computer system, wherein the second header is contiguous with the first header;

invoking a receive procedure of a second protocol layer of the communications protocol;

storing a start pointer and an end pointer to the payload data;

storing a second start pointer pointing to a first byte of the second header in the common data buffer;

adjusting the start pointer to point to the first byte of the first header according to the second protocol layer;

invoking a receive procedure of a first and higher protocol layer of the communications protocol; and

transferring to the first protocol layer the start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

25. (Previously presented) The computer readable medium of claim 24 wherein the program instructions remove a checksum added by the sending computer system from the received payload data in the common data buffer.

26. (Previously presented) The computer readable medium of claim 25 wherein the program instructions remove the checksum by adjusting the start pointer of the common data buffer to point to a memory location following the checksum.

27. (Previously presented) The computer readable medium of claim 24 wherein the program instructions transfer any application data or information required by the receive procedure of the first and higher protocol layer.

28. (Canceled)

29. (Previously presented) A method for processing payload data in a computer system using layers of a network communications protocol, the method comprising the steps of:

(a) storing the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header;

(b) processing the payload data using a first protocol layer of the network communications protocol; and

(c) processing the payload data using a second protocol layer of the network communications protocol, wherein the payload data is not copied during and between being processed by the first and second protocol layers.

30. (Previously presented) The method of claim 29 wherein the payload data does not move within the data buffer during and between being processed by the first and second protocol layers.

31. (Previously presented) The method of claim 29 wherein the processing step(a) further comprises the steps of:

(a2) positioning a first pointer to point to a first byte of the payload data; and

(a3) positioning a second pointer to point to a last byte of the payload data, wherein the first protocol layer uses the first and second pointers to locate the payload data for processing.

32. (Previously presented) The method of claim 31 wherein the processing step (b) further comprises the steps of:

(b2) adding a first element to the payload data; and

(b3) moving the second pointer to point to a last byte of the first element, wherein the first pointer does not move when the first element is added.

33. (Previously presented) The method of claim 32 wherein the processing step (c) further comprises the steps of:

(c2) adding a second element to the payload data; and

(c3) moving the second pointer to point to a last byte of the second element, wherein the second protocol layer uses the first and second pointers to locate the payload data and the first element to add the second element, wherein the first pointer does not move when the second element is added.

34. (Previously presented) The method of claim 33 wherein the first element comprises a header associated with the first protocol layer.

35. (Previously presented) The method of claim 33 wherein the first element comprises a checksum associated with the first protocol layer.

36. (Previously presented) The method of claim 33 wherein the second element comprises a header associated with the second protocol layer.

37. (Previously presented) The method of claim 33 wherein the second element comprises a checksum associated with the second protocol layer.

38. (Previously presented) A system for processing payload data using layers of a network communications protocol, the system comprising:

a processor for processing data from an application program; and

a component that stores the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header, wherein the component further processes the payload data using a first protocol layer of the network communications protocol and a second protocol layer of the network communications protocol, and wherein the payload data is not copied during and between being processed by the first and second protocol layers.

39. (Previously presented) A computer readable medium containing a computer program for processing payload data using layers of a network communications protocol, the computer program comprising program instructions for:

storing the payload data, a first header, and a second header in a data buffer, wherein the second header is contiguous with the first header;

processing the payload data using a first protocol layer of the network communications

protocol; and

processing the payload data using a second protocol layer of the network communications protocol, wherein the payload data is not copied during and between being processed by the first and second protocol layers.

40. (Previously presented) The computer readable medium of claim 39 wherein the payload data does not move within the data buffer during and between being processed by the first and second protocol layers.

41. (Previously presented) The computer readable medium of claim 39 wherein the program instructions for storing the payload data in a data buffer further comprise program instructions for:

positioning a first pointer to point to a first byte of the payload data; and

positioning a second pointer to point to a last byte of the payload data, wherein the first protocol layer uses the first and second pointers to locate the payload data for processing.

42. (Previously presented) The computer readable medium of claim 41 wherein the program instructions for processing the payload data using a first protocol layer further comprise program instructions for:

adding a first element to the payload data; and

moving the second pointer to point to a last byte of the first element, wherein the first pointer does not move when the first element is added.

APPENDIX B

EVIDENCE

(NONE)

APPENDIX C

RELATED PROCEEDINGS

(NONE)